

**Incorporating Intelligent Computer
Generated Forces in a
Human-in-the-Loop Simulation:
Phase One - Establishment of a
Research Testbed**

Peter Clark, Helen Pongracic,
Martin Cross, Jim Morton, and
John Fulton

DSTO-TR-0889

DISTRIBUTION STATEMENT A

Approved for Public Release
Distribution Unlimited

Incorporating Intelligent Computer Generated Forces in a Human-in-the-Loop Simulation: Phase One - Establishment of a Research Testbed

*Peter Clark, Helen Pongracic, Martin Cross,
Jim Morton, and John Fulton*

**Air Operations Division
Aeronautical and Maritime Research Laboratory**

DSTO-TR-0889

ABSTRACT

The Air Operations Division (AOD) of the Defence Science and Technology Organisation (DSTO) has developed a range of simulation tools whose capabilities include air combat modelling. The Air Operations Simulation Centre (AOSC) provides a facility for Human-in-the-Loop (HiL) simulation, and the BattleModel provides intelligent Computer Generated Forces (CGFs). The AOSC and the BattleModel have been integrated to create a research testbed for air combat simulation as Phase One of a Technology Demonstrator. This has been the first integration of operationally credible CGFs with a HiL facility in an air environment by DSTO. The CGFs have been implemented using dMARSTTM agent controlled fighter aircraft. Interaction with the AOSC has been achieved using Distributed Interactive Simulation (DIS). This paper outlines the rationale for using intelligent CGFs and HiLs in the same synthetic environment, details the development of the research testbed, describes the initial results achieved and makes suggestions for the way forward.

RELEASE LIMITATION

Approved for public release

20000127 127

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION

DSTO

Published by

*DSTO Aeronautical and Maritime Research Laboratory
PO Box 4331
Melbourne Victoria 3001 Australia*

*Telephone: (03) 9626 7000
Fax: (03) 9626 7999
© Commonwealth of Australia 1999
AR-011-122
October 1999*

APPROVED FOR PUBLIC RELEASE

Incorporating Intelligent Computer Generated Forces in a Human-in-the-Loop Simulation: Phase One - Establishment of a Research Testbed

Executive Summary

The Air Operations Division (AOD) of the Defence Science and Technology Organisation (DSTO) has developed a range of simulation tools whose capabilities include air combat modelling. The Air Operations Simulation Centre (AOSC) provides a facility for Human-in-the-Loop (HiL) simulation, and the BattleModel provides intelligent Computer Generated Forces (CGFs). The AOSC and the BattleModel have been integrated to create a research testbed for air combat simulation as Phase One of a Technology Demonstrator.

CGFs are a cost-effective way of providing extra players in a synthetic environment containing human participants. They are potentially a viable alternative in training, research of individual and team tactics, and research into human performance factors. Employing computer generated pilots whose behaviours emulate those of pilots with varying levels of experience would overcome a common problem (in both research and training) of obtaining experienced operational pilots, whose availability is both limited and expensive.

Phase One of the Technology Demonstrator has been the establishment of a research testbed, in which the BattleModel and the AOSC have been successfully linked and Entity State PDUs exchanged. Connectivity of the AOSC and the BattleModel with a simple scenario and simple tactics was achieved using DIS networking protocols. This demonstrated that CGFs and a HiL can, and do, interact in a real-time combat situation.

The demonstration scenario evolved as expected with orange and blue CGFs engaging each other and the HiL. However, the CGFs didn't behave as intelligently as they could have, as they were only endowed with limited intelligence. Several shortcomings of the system have been identified and these issues will be addressed in the next stage development.

This research testbed can now be used for further investigations of CGFs (via the BattleModel) and a HiL facility (for example, the AOSC) in the same synthetic environment. This will enable the construction of a full Technology Demonstrator which will demonstrate its usefulness by providing a source of suitable, intelligent CGFs to interact with Human-in-the-Loop facilities.

Further experimentation is needed to determine the realism of the interactions of HiLs and CGFs in air engagements, and from this to determine how best to utilise this capability. More complex scenarios are required to test both BattleModel tactics and human responses. To achieve this, further development is required to produce more realistic models (physical (e.g. missile flyout), tactical (e.g. teamwork), reasoning, and interface). dMARSTTM agents have been used as the CGFs and some development of the agents is required to provide more realistic HiL-CGF interactions.

Utilisation of the High Level Architecture (HLA) and dGame will be explored at a later development stage. The experience gained in linking the BattleModel with the AOSC will allow similar linkages to dGame, perhaps using HLA.

Authors



Dr Peter Clark
Air Operations Division

Dr Peter Clark graduated from the Australian National University (ANU) in Canberra with a BSc(Hons) in 1978. He was awarded a PhD in Physics from the ANU in 1981. He joined the Australian Defence Department in 1982 and initially worked in the field of operations analysis. In 1989 he was appointed for a five year period as the Scientific Adviser to Army's HQ Training Command in Sydney. In 1994 he was appointed as a Senior Research Scientist with the Air Operations Division of the Defence Science and Technology Organisation, based in Melbourne, where he has worked for the past six years on a range of topics involving simulation research. Dr Clark served a five year term (1993-1997) as Panel Chairman of the TTCP Technical Panel on Training Technology (HUM-TP2). Dr Clark has held the appointment of Australian Co-chairman of the International Simulation Advisory Group (ISAG) since 1997.



Dr Helen Pongracic
Air Operations Division

Dr Helen Pongracic graduated from Monash University with a BSc(Hons) in 1983 and was awarded a PhD in Mathematics from Monash University in 1988. She held several Postdoctoral Research positions in the U.K. working in areas of star formation and numerical astrophysics. In 1992, Helen took up a position as a Research Fellow at Sydney University to continue with this and other theoretical astrophysics research. Helen joined the Air Operations Division of DSTO in late 1995, where she has conducted research on a range of topics in the field of simulation.



Martin Cross
Air Operations Division

Martin Cross graduated from the Royal Melbourne Institute of Technology (RMIT) with a BEng (Aero) (Hons) in 1991. He conducted research for a Master of Aerospace Engineering in 1992/93 at RMIT. He joined the Air Operations Division of DSTO in 1994 and has worked primarily on the operational modelling of aircraft and their subsystems, and on the modelling and representation of the human reasoning components of AOD's air combat simulations SWARMM and BattleModel.

Dr Jim Morton
Air Operations Division

Dr Jim Morton was awarded a PhD in Physics from the Australian National University in 1997. Prior to his PhD studies he worked for ten years as a systems programmer with the Department of Social Security. He joined the Maritime Operations Division of DSTO in 1995, where he has worked on projects utilising Distributed Interactive Simulation. In 1998 he was attached to the Air Operations Division. Jim retired from DSTO in June 1999.



John Fulton
Air Operations Division

John Fulton graduated from the La Trobe University with a double degree of B.Sc (Computer) (Hons) and B.Eng (Electronic) (Hons) in 1995. He has worked in the Air Operations Division of DSTO since 1995, on real time simulation projects including developing infrastructure code for the Air Operations Simulation Centre (AOSC) and wide area networking of DIS protocols using ModSAF.

Contents

1. INTRODUCTION.....	1
2. USE OF CGFS AND HIL	2
3. BATTLEMODEL	3
3.1 Structure of the BattleModel	3
4. AIR OPERATIONS SIMULATION CENTRE.....	5
5. LINKING THE BATTLEMODEL AND THE AOSC	6
6. DEMONSTRATION SCENARIO.....	7
7. THE MODELS	9
7.1 Interface Models.....	9
7.1.1 BMtoAOSC Interface Model	9
7.1.2 AOSCtoBM Interface Model	10
7.2 Reasoning Models	10
7.2.1 BattleModel Aircraft Tactics – Orange Side.....	11
7.2.2 BattleModel Aircraft Tactics – Blue Side.....	11
8. AOSC CONFIGURATION	11
8.1 AOSC Hardware Configuration.....	11
8.2 AOSC Software Configuration	13
8.2.1 F/A-18 Cockpit and SGI interface	14
8.2.2 Spaceball	14
8.2.3 F/A-18 HOTASTA-NATC Flight Dynamics Model	15
8.2.4 F/A-18 instruments.....	15
8.2.5 Tactical Environment and DIS interface	15
8.2.6 Vega Image Generator	16
9. LIMITATIONS	16
9.1 Interface Models.....	16
9.2 BattleModel Tactics	17
10. RESULTS AND DISCUSSION	17
10.1 Execution of Experiment.....	17
10.2 How Intelligent were the "Intelligent CGFs"?	18
10.3 Missiles and Terrain.....	18
10.4 PDU Traffic	19
10.5 Benefits of Implementing Intelligent CGFs	20
11. FUTURE WORK.....	21
11.1 Immediate Development Work.....	21
11.2 Phased Plan for Scenario Development	22

11.3	Model Development.....	24
11.3.1	Reasoning Models (Agent Development).....	24
11.3.2	Interface Models	24
11.3.3	Physical Models	25
11.4	Future Development Work	25
12.	CONCLUSIONS.....	26
13.	REFERENCES	26
14.	ACKNOWLEDGEMENTS	28
	APPENDIX A:BATTLEMODEL DIS INTERFACE PROGRAMS.....	29
A.1.	Introduction	29
A.2.	Program Description	29
A.2.1	Overview	29
A.2.2	Receive_PDUs	30
A.2.3	PackageAndSend_PDUs.....	31
A.2.4	Flags used to pass status information within programs	32
A.2.5	Synchronisation when PDUs are out of order.....	33

1. Introduction

Computer Generated Forces (CGFs) are a cost-effective way of providing extra players in a synthetic environment containing human participants. They are a viable alternative in training, research of individual and team tactics, and research into human performance factors. Tactical air missions, where single or multiple pilots (either onside or opposing) fly against teams of intelligent computer generated opponents, can be simulated.

Faster and more sophisticated computing, networking and communications together with developments in Artificial Intelligence (AI) technologies enable the implementation of intelligent CGFs and virtual environments. This allows more realistic scenarios incorporating CGFs which interact with human participants, each other, and the virtual environment, and which can be played out in real time.

Distributed Interactive Simulation (DIS) [1] is an international standard (IEEE) networking architecture which allows human players and computer simulated entities (i.e. CGFs) to participate in the same exercise. Virtual environments, populated with both human and computer generated entities, enable the construction of large exercises conducted in a synthetic environment.

The BattleModel is a simulation architecture which incorporates intelligent computer generated forces. The intelligence component arises through the use of Agents incorporating artificial intelligence via dMARSTM (distributed Multi Agent Reasoning System¹) based on a beliefs, desires, and intentions framework.

A Technology Demonstrator is currently being developed which will link the BattleModel with a Human-in-the-Loop facility. Such a facility is the Air Operations Simulation Centre (AOSC) located in the Air Operations Division (AOD) of DSTO. Linking the BattleModel with AOSC stations and flying desks allows both Beyond Visual Range (BVR) and Within Visual Range (WVR) simulations, thereby providing a source of suitable, intelligent CGFs for the facility. This capability had previously been identified [2] as having potential to reduce the costs associated with setting up complex tactical environments, and will be of benefit in the areas of tactics development, training, evaluation and analysis.

Phase One of the development of this Technology Demonstrator has been the development of a testbed. A research testbed now exists which allows further investigation of the interaction of Computer Generated Forces (via BattleModel) with a Human-in-the-Loop (via the AOSC) in the same environment. This report details the development of the research testbed, describes the initial results achieved and makes suggestions for the way forward.

¹ dMARSTM is proprietary software developed by the Australian Artificial Intelligence Institute (AAIL).

2. Use of CGFs and HiL

There are two main reasons for using CGFs and HiLs in the same environment. The first is based on training and training research issues; the second is for analysis.

Training research involves studies of training effectiveness. Large scale training exercises using simulation facilities are often very costly and time consuming because of the requirement for a large number of humans, either as direct participants, or as computer force controllers. The use of *intelligent* CGFs in these evaluations would significantly reduce the required human (and consequently equipment and financial) resources.

Tactics development may also involve a large number of players. The development of tactics for large teams is complex and can be very time consuming. In many cases the requirement for a large number of human participants may render the simulation of complex tactics uneconomical or impractical, from both a human resource availability, and a technical, point of view. The use of CGFs for some, or the majority of players, during tactics development eliminates the requirement for a large number of human players. This reduces the cost of the development, and increases the repeatability of simulations.

Knowledge acquisition from domain experts may involve an expert (i.e. a pilot) sitting in a simulation environment and flying against computer generated forces. The expert will be able to verify the behaviour of the CGFs and improve the credibility and operational accuracy of the models, and will also understand the required behaviour and actions of the models. Using realistic CGFs the expert could also be observed flying "real" missions to allow easier and more accurate transfer of domain knowledge.

Human factors research (such as display design and workload analysis) may involve a domain expert (e.g. a pilot) flying a "real" mission. Intelligent CGFs can be an inexpensive source of other players in the scenario. Because the CGFs behave the same way given the same circumstances, it is possible to conduct repeated and controlled experiments. Consequently, it is possible to develop metrics for measuring human and CGF performance.

Flying against different arrangements of CGFs in experimental scenarios enables the capture of novel tactics and innovative behaviour from human experts when testing or developing new technologies or systems.

The ability to simulate team versus team missions in a distributed real-time computing environment is a primary requirement for many applications. Using CGFs and Humans-in-the-Loop in the same simulation reduces the requirement for expert pilots and equipment for training and tactics development. Mission rehearsal is a specific form of training, and through the use of CGFs can be conducted at any time regardless

of weather conditions, equipment or pilot availability. In addition, such use of CGFs may also reduce the number of instructors required, some of whom may not always be available at appropriate times.

In the analysis domain, adding a HiL simulation to a CGF driven operations analysis provides a new method for the evaluation of total systems. Different system configurations could be trialled and the effectiveness of the configuration evaluated. For example, a human operator could be placed in a CGF role, such as a fighter-controller in an AEW&C system, and experiments can be run which determine whether any unexpected consequences occur. Simulations combining CGFs and HiLs provide repeatability (control of the intelligent CGFs), variability (from the unexpected inputs of the HiL), validation (credibility of agent tactics and behaviours which can be evaluated by expert human pilots), and traceability (of the decision making process).

3. BattleModel

The BattleModel is a simulation environment developed by the Air Operations Division (AOD) of DSTO and is currently used for the simulation of both small-scale and large-scale air combat missions for operational analysis studies [3]. The BattleModel architecture provides a framework for simulations. It controls the routing of data from the various models (physical, tactics, reasoning), and the timing and updating of the models. The architecture provides a "plug and play" capability which allows the use of established and verified models in the scenarios.

The BattleModel builds on the experience of AOD's engagement and mission level air combat simulator SWARMM [4]. As a result of this evolution, the models currently used with the BattleModel architecture are operationally valid and verified physical and engagement models. The BattleModel is well suited to Australian Defence as it contains validated Australian tactics and doctrine. A range of pilot behaviours is currently available, allowing the BattleModel to represent computer generated entities with differing levels of experience and ability.

The tactics used within the BattleModel have been developed over a period of time in response to RAAF tasking to investigate various aspects of tactics of a number of platforms. The result is a suite of tactics representing a range of combat situations. The tactics are operationally credible as they are based on current RAAF tactics and have been used to provide tactical advice to operational RAAF squadrons.

3.1 Structure of the BattleModel

The BattleModel is run by a "battle manager" which controls the architecture, manages the scenario (by first loading and then initialising it) and manages the simulation (by

starting, pausing, resuming, stopping and restarting it). It is known to all models in the simulation.

Various services are available, including a model identity database; time services; information services; and measures of effectiveness, debug and output services (which provide mechanisms for data retrieval and analysis). These services are known to all models in the simulation.

For a particular scenario, the appropriate models (physical, tactical and reasoning) are selected and their required parameters specified. The BattleModel architecture initialises each of the models, and then controls the execution of the models during the simulation. This control can be categorised into two roles.

The first role is the control of the order of execution of the models. At each timestep a particular model may require data generated by another model at the same time step. The BattleModel architecture ensures that the order of execution is as required. This requirement is specified by the users. The BattleModel architecture is not distributed. In a distributed system this control of the order of execution of models is not always possible, whereas such control in the BattleModel ensures the use of data generated at the appropriate time step, rather than data that is outdated. A distributed system typically performs the same task by using dead-reckoning models.

The second role is the control of the distribution of data amongst the models. The BattleModel architecture uses a series of Information Servers to allow this distribution. A particular model registers as a client of any Information Servers from which it requires data, and registers as a server to any Information Servers to which it provides information. For example, a typical radar model would subscribe to a Geometry Information Server, which contains information about the positions of all platforms in the scenario, and would register as a server to a Radar Track Information Server, which would contain the radar tracks generated by the radar model. Another model, e.g. a radar display model, would register as a client of the Radar Track Information Server, and would use the track information to generate some sort of display of the tracks on the screen. More than one model can (and often does) register as a client to an Information Server and uses the information in the Information Server. The BattleModel architecture controls the sending and receiving of data to these Information Servers.

The BattleModel is a time-based simulation and can be run at real time or faster than real time. Human-in-the-Loop simulations require real time and this is catered for in the Time Server. The Time Server controls the updating of information and "ticking" of the various models. It supports multiple ticking rates as different models can, and do, run at different rates. For example, the missile model is updated 100 times per second, whereas a radar model may be updated 10 times a second. The models are ticked based on data dependency. The Time Server enables the functions "start", "pause", "stop" and "resume" to occur and knows about all running models.

Information Servers are data transportation channels between entities in the simulation. They link multiple servers and clients. Clients can specify a filter function, so that only relevant information is provided. Connection of models to servers is via a registration and subscription scheme. Registration informs Services of an interest, provides the initial connection to an Information Server and is performed by both servers and clients. Subscription informs a Service of a requirement, can only be used by clients, ensures data dependency and provides the filters.

4. Air Operations Simulation Centre

The Air Operations Simulation Centre (AOSC) provides a distributed, real-time, human-in-the-loop simulation environment for the conduct of research into the performance of aircrew and aircraft systems in operational scenarios. It also provides long term support to Australian Defence Force (ADF) operations and acquisitions, particularly, but not exclusively, in air related issues focussing on the importance of overall systems and the human dimension.

In addition, the facility is a focus within DSTO for exploiting rapid developments in simulation technology, such as DIS/HLA (Distributed Interactive Simulation / High Level Architecture) networking architectures, with various applications, including training and mission rehearsal. AOD has been acknowledged as DSTO's centre of excellence in matters concerning Distributed Interactive Simulation, and has contributed greatly by providing technical advice to the ADF's leading simulation networking project, Navy's Project SEA 1412 [5].

The AOSC is comprised of a series of simulation stations with various degrees of fidelity networked together in a simulated tactical environment providing flexible, reconfigurable, scenario-driven tools. Major equipment includes:

- A Partial Dome Display System (produced by SEOS Displays Ltd), which consists of a partial six-metre diameter dome with a field-of-view of 200 degrees in the horizontal by 100 degrees in the vertical. The visuals are provided by six Barco graphics projectors that typically operate at 1280 x 1024 pixels, with a refresh rate of 60Hz. Higher resolution and refresh rates are possible.
- Two Helmet Mounted Displays (HMDs) produced by Kaiser Electro Optics Inc. The field of view is 60 degrees per eye, with colour imagery, for one helmet, and 40 degrees per eye, with colour imagery, for the other helmet. Head tracking is incorporated.
- A Collimated Display consisting of a horizontally mounted video monitor which projects the image onto a collimating mirror (via a beam splitter) to give the

impression that the images are generated at infinity. The field of view is 53 degrees by 33 degrees with a maximum resolution of 1280x1024 pixels.

- A number of Simulation Cockpits including a medium fidelity F/A-18 cockpit, a medium fidelity F111 cockpit, and a medium fidelity BlackHawk/Seahawk cockpit. These cockpits can be used with either the Partial Dome Display System, with the HMDs, or with both (with HMD in see-through mode providing Head-up-Display information).
- Three additional team stations with lower-fidelity visual displays and flying-desk type cockpits.
- Two advanced SGI computer systems with image generators; one Onyx with sixteen R4400 type RISC processors, operating at 150MHz, and one Onyx2 with eight R10000 type processors operating at 200 MHz.
- The simulation software packages: STAGE (Scenario Toolkit and Generation Environment), MultiGen (a visual database designer), VR-Link (for DIS connectivity), Vega Image Generator, VAPS (for creating flight instrument displays), and ModSAF (Modular Semi-Automated Forces), as well as "IRIX" (the Silicon Graphics variant of UNIX), and "Performer" (the SGI visual application toolkit).

5. Linking the BattleModel and the AOSC

The BattleModel and the AOSC are linked using DIS protocols. DIS enables connectivity between crewed and automated simulation applications, so that real-time, Human-in-the-Loop, multiplayer scenarios are possible. Each simulation may operate on different remote computers, but all of them participate in the same virtual world. The DIS protocol is a set of standards for communicating information about a virtual world among the many participating applications.

The AOSC is already DIS compliant, so an interface was required to the BattleModel to ensure it could also communicate via DIS (see section 7.1). The HiL simulation (in the AOSC) sends DIS PDUs (Protocol Data Units) onto the network. These PDUs are picked up by the DIS interface to the BattleModel, which then incorporates the "extra" entity or entities in its calculations. The DIS interface to the BattleModel also produces DIS PDUs for all BattleModel entities (aircraft, missiles, etc.) and sends these onto the network to be picked up by the HiL.

The Virtual Reality networking toolkit (VR-Link), produced by M&K Technologies Inc., is a software product which provides the DIS networking communications and transfer protocols needed to create an interface between the Human-in-the-Loop station and the BattleModel. The VR-Link package also includes the "Stealth" observation platform

which provides a three dimensional out-of-the-window view of the virtual world, allowing the operator to attach the eye-point to various DIS entities in the scenario. Data loggers enable the recording and playback of a scenario. Other visualisation devices for viewing the simulation include: BattleView (a BattleModel viewer which displays a God's eye view); dMARS Viewer (tactics are highlighted as they are selected by the CGFs, providing traceability of the decision-making process); RSAD (an AOSC Rapid application Situation Awareness Display which can be used for debriefing and analysis); and the collimated display (which provides out-the-window visuals for the HiL during the simulation).

The majority of fighter tactics can be divided into either Beyond Visual Range (BVR) or Within Visual Range (WVR) tactics [2]. Technological developments in missiles, radar, and communication systems have led to increased interest in BVR tactics. From an economical point of view, BVR engagements have the potential of eliminating opponents without the high risk and high losses involved in WVR air-to-air combat. From a modelling point of view, BVR engagements are much less dynamic than WVR engagements. BVR air combat is more influenced by tactical selection and less by individual pilot flying skills, the reverse being true for WVR air combat.

Several Human-in-the-Loop stations are available within the AOSC. These have the ability to present high quality out-the-window visual scenes suited to WVR engagements. For BVR engagements, human pilots could participate using flying desks with the appropriate sensor displays, as high quality visuals would not be required.

Whilst the AOSC is already in use for various research activities, more could be achieved if intelligent and credible CGFs could interact with the various human stations. The AOSC is already DIS compliant, so a suitable interface between the BattleModel and the AOSC (based on the DIS protocol) is required to achieve this interaction. The development and testing of this interface has been one of the initial achievements of the Technology Demonstrator under development.

6. Demonstration Scenario

To demonstrate the interaction of a HiL simulation with intelligent CGFs, a relatively simple scenario was chosen. The demonstration scenario consists of five aircraft, four of which are BattleModel CGFs and one a HiL. The aircraft are on two sides (two CGFs and the human on the blue side and two CGFs on the orange side). The scenario is shown in Figure 6.1.

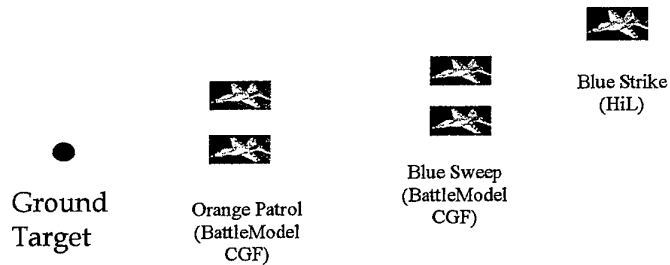


Figure 6.1: Demonstration Scenario

The orange side consists of two CGFs flying a patrol as a pair in a defensive barrier position in defence of a ground target. The blue side consists of three aircraft and is composed of two CGFs and one HiL. The HiL aircraft is flying in a strike role and its goal is to bomb the ground target. The two CGFs are flying as a pair in a sweep role. Their goal is to neutralise any enemy air defences to allow the strike aircraft to bomb the ground target.

The two orange aircraft flying patrol will do so until they detect enemy aircraft. Upon detection they will attempt to defend the ground target by performing a team intercept on the sweep aircraft. If they kill the blue sweep aircraft they will then intercept the strike (HiL) aircraft. The two sweep aircraft will perform a team intercept on the orange patrol aircraft when they are detected. The strike (HiL) aircraft will be attempting to bomb the ground target, but must also react to the orange patrol aircraft as appropriate.

The only non-BattleModel entities in the simulation are the single HiL aircraft and its associated missiles. All other aircraft and missiles are BattleModel objects. From a tactics point of view the CGFs are given a small number of simple goals and their tactics suite limited to allow clear observation of behaviour.

The demonstration scenario is complex enough to demonstrate the required behaviours, but at this stage does not include team interaction between CGFs and HiL pilots on the same side. Scenarios that contain these interactions will be addressed in future research. The issue of team tactics and the modelling of the relations both inside and amongst teams and organisations are complex and are a significant area of research in the modelling community [6, 7, 8]. There is ongoing development of fighter tactics in AOD and one of the primary aims is the development of team-related tactics [9]. The tactics developed in AOD can be readily used in BattleModel simulations.

7. The Models

For the development of the research testbed, several models were attached to the BattleModel architecture. The CGF fighters were modelled using three components; an aircraft platform model, a radar model, and a reasoning model. Two other models were developed as part of the BattleModel DIS interface; one to generate the DIS PDUs for entities within the BattleModel (called BMtoAOSC), and one to process incoming DIS PDUs (called AOSCtoBM).

The aircraft and radar models consist of legacy code and are validated and operationally credible physical models. The reasoning model used to represent the fighter tactics was implemented using dMARS™, a BDI (beliefs, desires and intentions) agent modelling system. A dMARS™ agent represents each of the fighter pilots. The DIS interface models were constructed specifically for Phase One of the Technology Demonstrator and are described below.

7.1 Interface Models

7.1.1 BMtoAOSC Interface Model

The role of this interface model is to obtain the data from the BattleModel pertaining to all the BattleModel generated aircraft and missiles, and to generate PDUs as required in order to send the data to the AOSC. The BMtoAOSC interface is comprised of VR-Link library routines and user defined code.

BattleModel Aircraft

Each BattleModel aircraft is represented within the BMtoAOSC interface by a data structure which defines position, velocity and orientation. At each simulation time the data in this structure is updated as new data are obtained from the BattleModel.

The BMtoAOSC interface sends *Entity State PDUs* using these data as required. The frequency of PDU generation depends on several factors, but primarily the dead reckoning model used by the DIS interface. The generation of PDUs is independent of the BattleModel architecture and is purely a property of the VR-Link utilities implemented in the interface.

BattleModel Missiles

Each BattleModel missile is represented within the BMtoAOSC interface by a data structure which defines position, velocity and orientation. At each simulation time, new data are obtained from the BattleModel. The BMtoAOSC interface constructs and sends *Entity State PDUs* using this data as required.

When a missile is launched a *Fire PDU* is generated. The BMtoAOSC interface needs to know when a missile is launched and the DIS-identification of the launching aircraft. When the BattleModel determines that its missile hits the HiL aircraft, a *Detonate PDU*

is sent by the BMtoAOSC interface. This will inform the AOSC that a BattleModel missile has detonated. The HiL will then take appropriate action (via the DIS protocol) to remove itself from the simulation. A final *Entity State PDU* is sent at the same time as the *Detonate PDU*.

7.1.2 AOSCtoBM Interface Model

The role of this interface is to obtain the data from the HiL aircraft (via DIS PDUs) and then send the relevant data describing this aircraft and its missiles to the BattleModel. From the BattleModel point of view, the HiL aircraft is just another aircraft model.

The HiL aircraft and each of the missiles on the HiL aircraft are defined in the AOSCtoBM interface. A DIS entity identification is obtained from the initial *Entity State PDU* for each remote entity (aircraft and each missile). Each remote entity is a separate model from the BattleModel point of view. The AOSCtoBM interface obtains data from incoming PDUs generated by the HiL facility, stores the required data locally, then converts and sends the data to the BattleModel in an appropriate format.

HiL Aircraft

The HiL aircraft is represented within the AOSCtoBM model by a data structure which defines position, velocity and orientation. At each simulation time the data in this structure are updated with new data obtained from the *Entity State PDUs* for the aircraft or dead-reckoned data. The updated data are then sent to the BattleModel.

HiL Aircraft Missile Firing

When the HiL facility fires a missile a *Fire PDU* is generated, and processed by the AOSCtoBM interface. The missiles are represented within the AOSCtoBM interface by a data structure which defines position, velocity and orientation. At each simulation time the data in this structure are updated using new data obtained from the *Entity State PDUs* for the missiles or dead-reckoned data. The updated data are then sent to the BattleModel.

To simulate the detonation of a missile fired from the HiL aircraft, a *Detonate PDU* is issued by the HiL simulation. This is received by the AOSCtoBM interface and the BattleModel informed that the missile identified in the *Detonate PDU* has detonated. The BattleModel then takes appropriate action in response to the detonation.

7.2 Reasoning Models

AOD has used dMARSTTM to model air combat tactical behaviour for several years [6, 8]. There is continuing development of F/A-18 fighter tactics (using dMARSTTM) within AOD, as well as some research into the use of intelligent agents in the air combat environment [10, 11, 12, 13]. As a result of this work there is a large suite of tactics available which can be used with aircraft in BattleModel scenarios. A simplified selection of these tactics was used for this research testbed. These are described below.

7.2.1 BattleModel Aircraft Tactics – Orange Side

BattleModel tactics for the orange side are as follows. The orange aircraft are initially flying a pre-defined patrol in an area between the blue target and the direction of the incoming blue force. The leader and the wingman are both searching for bandit aircraft.

When the orange aircraft detect the blue sweep aircraft they sort the targets, then perform either a single side intercept or a pincer intercept. The selection of intercept is based on the bearing of the blue sweepers when the orange patrol detects them. If the magnitude of the difference between the bearing to the blue sweep aircraft and the heading of the orange aircraft is less than 20 degrees, a pincer intercept will be performed; if it is greater than 20 degrees, a single side intercept will be performed.

The orange fighters each try to kill their targets. When one of the orange aircraft achieves this, it then attacks the other blue sweep aircraft if it (blue sweep) has not been fired at. If both blue sweeps are destroyed, the orange fighters will return to their patrol role and resume searching for contacts. If the orange aircraft detect the blue strike aircraft they perform an intercept as described above.

7.2.2 BattleModel Aircraft Tactics – Blue Side

BattleModel tactics for the blue side are as follows. The leader of the blue aircraft sweep pair is initially flying a heading towards the Strike's target, and the wingman is flying formation. When the aircraft detect the orange patrol they sort the targets, and then perform either a single side intercept or a pincer intercept using the same criteria as the orange fighters.

The blue sweep fighters each try to kill their targets. When one of the blue aircraft kills its target it then attacks the other orange aircraft if it (orange) has not been fired at or killed. If both orange sweepers have been destroyed, the blue fighters return to flying towards the Strike's target.

8. AOSC Configuration

8.1 AOSC Hardware Configuration

For the research testbed stage of this Technology Demonstrator, the F/A-18 cockpit was used in conjunction with a collimated display. The F/A-18 cockpit (Figure 8-1) has VGA resolution video displays for cockpit instruments. A flight dynamic and aviation model is used to simulate most aspects of the F/A-18 at the 87X version level. This model simulates Head-Up-Display (HUD), other avionics displays and most effects due to cockpit levers and switches. The throttle and stick are spares from an actual

F/A-18. Flight controls, sticks, throttles, pedals etc. have been fitted with transducers and their electrical outputs interfaced to a dedicated Motorola data acquisition system. The Motorola system transfers the cockpit information to the AOSC Silicon Graphics Incorporated (SGI) Onyx computers via a reflective memory bus.

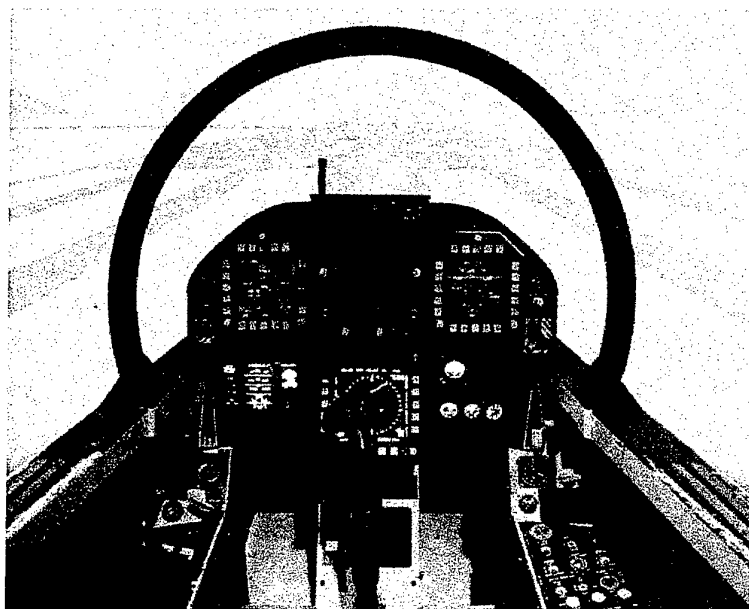


Figure 8-1: The AOSC F/A-18 medium fidelity cockpit.

The collimated display was one of the original display devices used within AOD prior to the construction of the AOSC, and is still a useful display device. The collimated display (Figure 8-2) consists of a monitor mounted on top, facing down, with the light deflected ninety degrees to the rear via a half-reflecting, half-transmitting mirror (known as a beam splitter). The light is then totally reflected in a curved mirror at the back (approximately 36 cm wide by 24cm high), back through the beam-splitter to the view of the pilot.

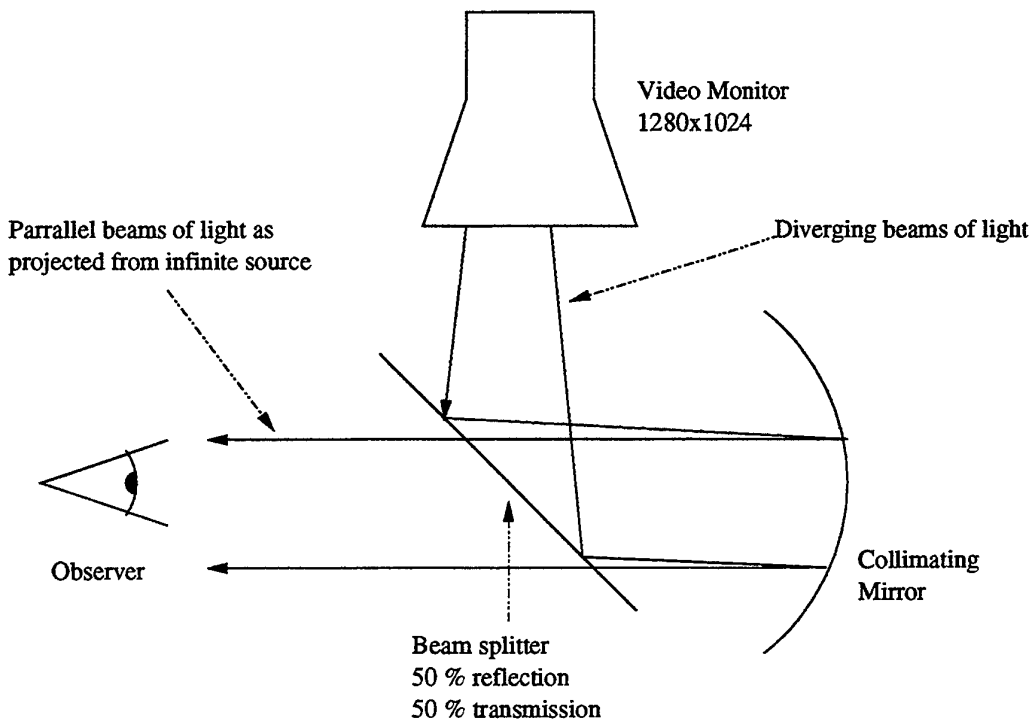


Figure 8-2: Operation of Collimated Display

8.2 AOSC Software Configuration

The AOSC has a selection of software modules that can be configured to support different types of experiments. The software modules used in Phase One of the Technology Demonstrator in the AOSC were configured as shown in Figure 8-3.

The software in the AOSC can be broken into two separate classes: asynchronous and synchronous. Asynchronous modules are used to connect non-AOSC software (such as the Vega Image Generator and DIS network traffic) and real world inputs (such as pilot stick movements) to the synchronous modules. Asynchronous modules generally have a synchronous module to interface them to the rest of the system. The synchronous modules exchange variables at the end of their execution cycle. This exchange is controlled so that it occurs at the correct execution rate of the module and occurs in a repeatable fashion. All data variables exchanged between synchronous modules can be recorded for later analysis.

The following sub-sections outline the use and configuration of each software module shown in Figure 8-3.

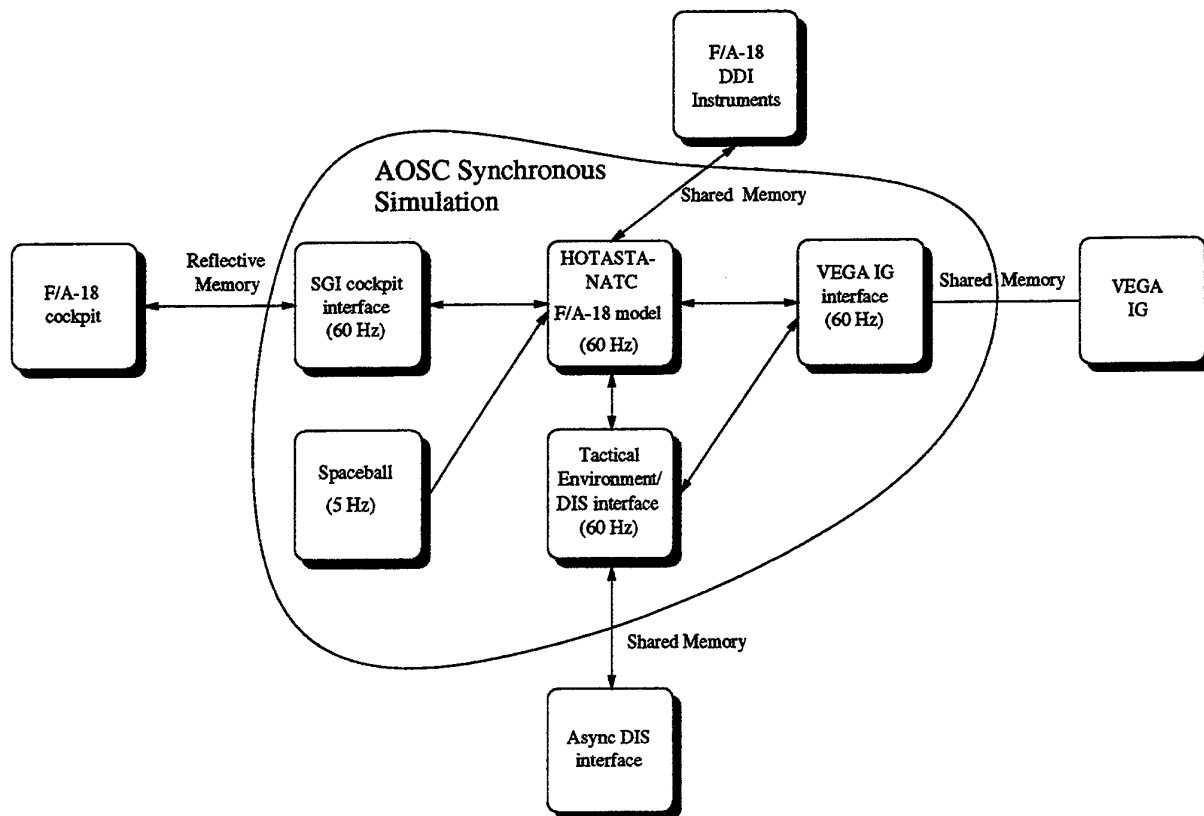


Figure 8-3: Software Configuration of the AOSC for BattleModel Integration

8.2.1 F/A-18 Cockpit and SGI interface

The F/A-18 cockpit software runs on a dedicated Motorola data acquisition system. The Motorola system is responsible for sampling the analog inputs (e.g. stick and throttle), digital inputs (e.g. switch), and turning on output devices (e.g. lights and the up-front controller output). All data to and from the Motorola is transferred via a reflective memory bus to the SGI computers.

The SGI interface simply reads and writes the inputs and outputs to and from the reflective memory bus. It provides an interface for the rest of the AOSC synchronised simulation to read and write to the cockpit controls and displays.

8.2.2 Spaceball

The Spaceball is a six-degrees-of-freedom device used to move around in 3-dimensional space. The Spaceball has ten push buttons, which can be used to control parts of the simulation. In this exercise the Spaceball was used to position the F/A-18

model in the virtual world. One of the push buttons was configured as a toggle to either:

- reset the F/A-18 aerodynamics model and allow the model be positioned using the Spaceball; or
- allow normal flight of the aerodynamics model.

The Spaceball was run at a lower execution rate of 5 Hz as its output variables were not required at a high rate to position the F/A-18 model.

8.2.3 F/A-18 HOTASTA-NATC Flight Dynamics Model

The Hands On Throttle And Stick Training Aid (HOTASTA) is a model of the F/A-18 avionics and radar. This model implements the 87X version of the avionics. AOSC staff have integrated the HOTASTA model with a flight dynamics model obtained from the Naval Air Warfare Center (formerly Naval Air Training Center (NATC)). The HOTASTA model consists of 100,000 lines of Fortran and C. The NATC model is about 30,000 lines of Fortran and C. The model was configured to simulate a gross weight of 32000 lb and 6000 lb of fuel.

8.2.4 F/A-18 instruments

The F/A-18 instrument software implements the F/A-18 displays at the 87X version level. The displays available include: the Head Up Display (HUD), left and right Digital Display Interfaces (DDIs) and Horizontal Situation Interface (HSI). The code is written in IRIS-GL and reads all display information through a shared memory interface from the HOTASTA model.

As the HUD device was not fitted to the F/A-18 cockpit, the left DDI showed the HUD output. The right DDI showed the radar interface.

8.2.5 Tactical Environment and DIS interface

The AOSC Tactical Environment is a database of the entities participating in the exercise. The Tactical Environment contains all known information about each entity, including platform type, position, velocities, electronic warfare status (radars, jammers) and appearance attributes (smoking, on fire, etc.). The number of entities in the Tactical Environment is set at compile time and was set to 20 entities (1 internal and 19 external) for this exercise.

The DIS interface talks to the Tactical Environment via a shared memory interface. It outputs all entities modelled by the AOSC and updates the Tactical Environment with all external DIS entities. If there are more entities in the DIS world than is available in the Tactical Environment, the DIS interface will select the closest entities for inclusion in the Tactical Environment.

8.2.6 Vega Image Generator

The Vega Image Generator is used to show the out-the-window scene for the F/A-18 pilot. The Vega Image Generator is a third party product from MultiGen-Paradigm² that has been enhanced by the AOSC to meet the special requirements of the AOSC. The Image Generator is capable of modelling Infrared (IR) and FLIR (Forward Looking Infrared) imagery, as well as normal day- and night-time scenes. The interface to the VEGA Image Generator was written by AOSC staff and is used to control what is displayed by the Image Generator. These include the viewer's eye point, other entity positions, time of day, and weather effects.

The database used in this exercise was the "sea plate with isle". A picture of the island from this database is shown below. The target for the strike aircraft in the trial scenario was on this island.

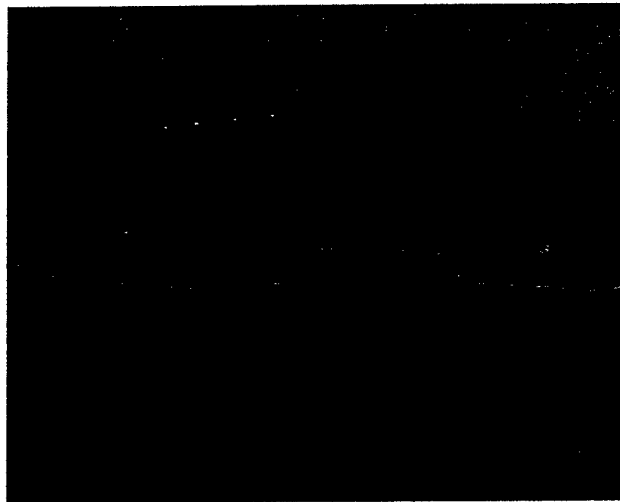


Figure 8-4: "Sea plate with isle" database

9. Limitations

9.1 Interface Models

Currently, BattleModel Information Servers do not contain data which can identify the source of the simulation entity (HiL facility or BattleModel). This causes a problem with the BMtoAOSC interface because only *Entity State PDUs* for BattleModel entities are required to be generated by this interface model. The current solution is to classify

² www.paradigmsim.com

all non-BattleModel entities as having the same BattleModel parent. This allows the extraction of data from the BattleModel Information Servers for all but the parent (the HiL aircraft).

This solution is acceptable for a small number of non-BattleModel entities (one aircraft and a few missiles), but would be impractical for a larger number of entities. The solution is to create a DIS Information Server for the BattleModel. This Server must contain data describing all non-BattleModel aircraft and missiles in the scenario, and must also know the DIS and BattleModel identifications for each of the entities. The DIS Information Server will be necessary if more than a small number of non-BattleModel entities are required in an exercise.

9.2 BattleModel Tactics

The tactics implemented in the demonstration exercise were deliberately chosen to be relatively simple in order to demonstrate the feasibility of the system. The functionality of the tactics used is as described in Section 7.2. A large suite of tactics has been developed by AOD for a range of aircraft types, acting in a range of roles. These are available and, with a small amount of development, could be used for future scenarios and exercises.

10. Results and Discussion

The experimental testbed was established and several experiments successfully carried out. The set up was as described in section 6. The major result of the experiments was that the CGFs and HiL could detect and react to each other in the same virtual world. The HiL cockpit appeared in the BattleModel 2D display and was recognised by the BattleModel. The BattleModel generated CGFs appeared in the AOSC display. Tactics appeared to be as expected with orange and blue CGFs engaging each other and the HiL [14].

10.1 Execution of Experiment

The following description outlines activity within the final experimental run. The blue sweep aircraft headed towards the target until they were detected by the orange patrol aircraft. The leader of the orange patrol aircraft detected the leader of the blue sweep. This was evidenced by orange leader's change of direction to follow the blue leader. When the blue sweep leader was within range the orange leader shot him down. Blue sweep wingman then locked onto orange patrol leader and shot him down. Orange patrol wingman then detected the Strike aircraft (the HiL) and turned to follow him. The HiL manoeuvred and managed to evade the orange patrol wingman. The orange patrol wingman then pursued the blue sweep wingman away from the target allowing the HiL strike aircraft to successfully reach the target, thus achieving its mission.

10.2 How Intelligent were the "Intelligent CGFs"?

The behaviour of the CGF entities was as expected with occurrences of detection, tracking, recognition as friend or foe, attack and evasion, and kill. In this particular example, one blue sweep was shot down and one orange patrol shot down. However, there were several areas which didn't show the CGFs behaving as intelligently as they could have, as they were only endowed with limited intelligence. These are illustrated by:

- Once a CGF had radar lock on another entity, it wasn't able to locate any other entity even if the first one it locked onto was killed. The CGF continued along the vector of the last known contact; away from the area of action. This should be able to be rectified by producing more sophisticated tactics for the chase and detect portion of the CGF tactics.
- Currently, the CGFs are endowed with a constant velocity, which inhibits their ability to chase the HiL (which can vary its speed). It was relatively easy for the HiL to shake the CGF once it had been detected, and make its way to the target.
- In one experiment, the HiL was started in front of the blue CGF sweep pair, and initially made contact with the orange CGFs to draw them to the area of battle. This was because at one stage the CGFs had difficulty detecting the HiL - once a CGF had radar lock on another entity (usually another CGF) it was unable to detect any other aircraft.
- Once the orange CGF patrol had successfully shot down its opponents it was required to return to its patrol role. It didn't behave as expected in this area as it continued along the vector of the last known contact. This problem could be overcome with more sophisticated tactics for the CGFs.

10.3 Missiles and Terrain

The missile activity described in section 7.1 is a facility which is implemented in the BMtoAOSC and AOSCtoBM interfaces but was not utilised in the Phase One testbed development. The primary reason for this is that the BattleModel does not currently have a missile flyout capability, but instead uses a "probability of kill" to determine whether aircraft are shot down. A missile flyout model exists, which could, with a little work, be implemented in the BattleModel, but was not available for the testbed demonstrations. From the HiL aspect a missile model is essential to ensure credibility of the simulation. It is not very relevant to fly against the throw of a dice!

The interaction between the HiL and CGFs occurred over the sea. A reference point for the ground target was required, hence the visual database "sea plate with isle" was chosen. However, the issue of matching the visual database for the HiL and the non-terrain database for the CGFs needs to be addressed.

10.4 PDU Traffic

The logged exercise DIS PDUs were analysed using tools provided with the MÄK Technologies networking toolkit VR-Link and common UNIX utilities. VR-Link tools used were:

- LgrDump - which converts the contents of the binary DIS PDU logger file into ASCII format; and
- Buffgrep - which searches its input stream for text buffers containing a specified text string pattern. Buffers containing the matched pattern are sent to the output stream.

Standard Microsoft Excel functions and charting capabilities were used to provide statistical analysis and graphical output on the logged data files after transferring them from the UNIX environment to PC Windows.

Only Entity State PDUs were present in the initial experimental runs. The number of Entity State PDUs generated per second was converted into bits per second. This involved adding an associated overhead wrapper of 428 bits to the actual recorded PDU size, because the VR-Link analysis tools remove this overhead.

The Entity State PDU spectrum generated for the Human-in-the-Loop in terms of bits per second, is shown in Figure 10.1. The general trend of peaks and troughs depending on the activity of the HiL is as expected.

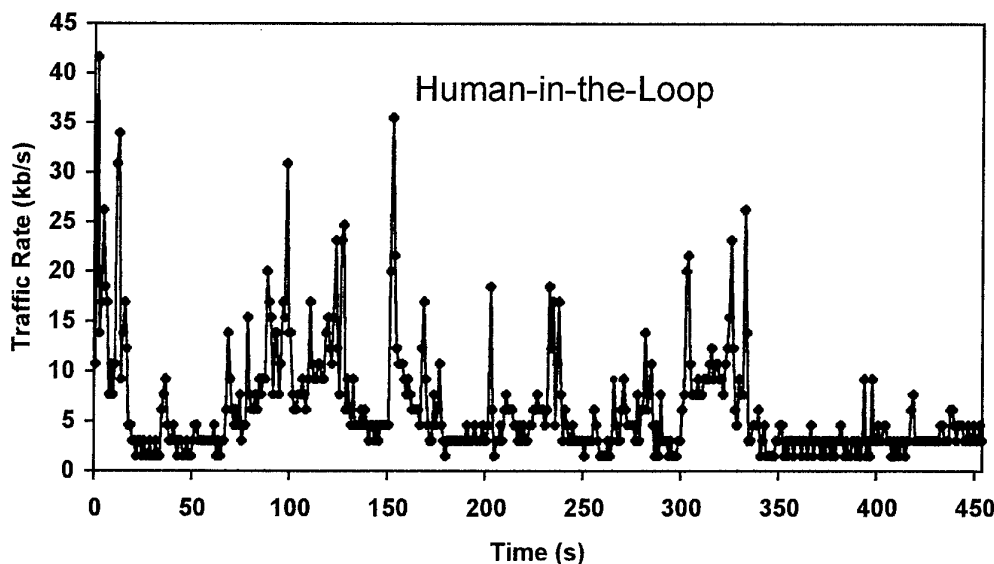


Figure 10.1 DIS traffic in kbits per second for the Human-in-the-Loop

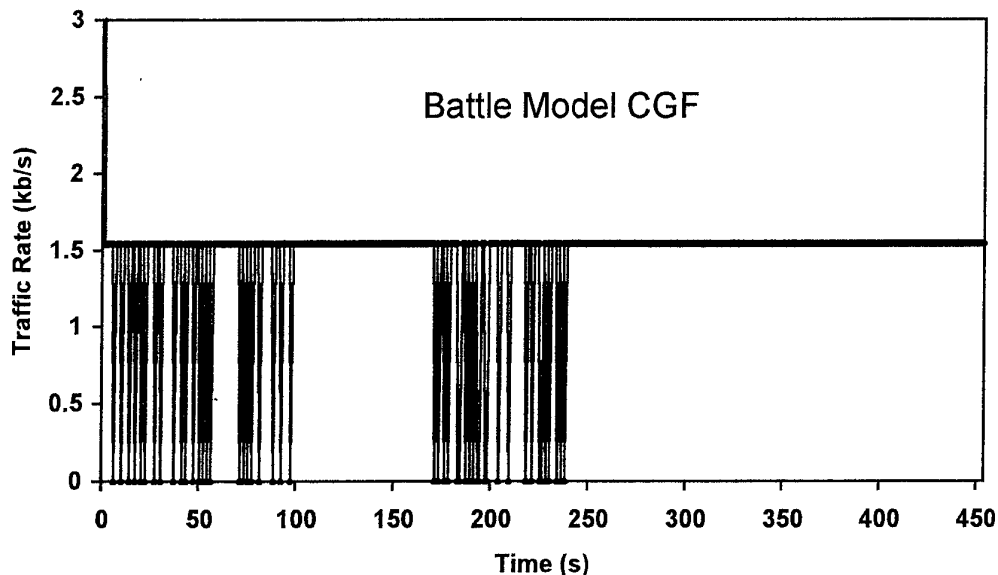


Figure 10.2 DIS traffic in kbits per second for the Orange 1 CGF.

The Entity State PDU spectrum generated for the orange CGF (patrol 1) in terms of bits per second, is shown in Figure 10.2. The spectrum is not what was expected, since other CGFs (generated by ModSAF for example) have shown the same structure as was found for the HiL in Figure 10.1. Further analysis indicated that a maximum of only one Entity State PDU was being generated by the BattleModel CGF per one second interval (at other times no PDUs were generated per second). Investigation showed that the sampling rate in the BattleModel DIS Interface programs was set too low. This will be rectified in the further development of the testbed, by using more realistic sampling intervals at the BattleModel DIS interface (for example, increasing the sampling for entity state PDUs from 1 per second to 20 per second).

10.5 Benefits of Implementing Intelligent CGFs

The inability of current CGFs to emulate the human decision making process leads to "reactive" behaviours, which are characterised by being predictable and scripted. Whilst such responses are more easily deceived, CGFs can be endowed with more realistic behaviour through the provision of both proactive and reactive responses. This can be achieved if the CGF is capable of "inferring the intent" of opposing entities in the simulation, and then developing counter-tactics which feature proactive behaviour [15].

CGFs, in attempting to emulate human behaviour, should reflect the variety of skill levels and experience typical of human pilots. Team and group leaders would be expected to make faster and more tactically relevant decisions than pilots just out of training. This would overcome the common problem, in research, training, and analysis, of obtaining experienced operational pilots, whose availability is both limited and expensive.

Human-in-the-Loop simulations have been used at almost all levels of conflict – engagement, mission, battle and campaign. In modern air warfare engagements the majority of operational interaction begins beyond visual range (BVR), with same-side pilots working cooperatively in teams and possibly in a complex electronic warfare (EW) environment. For BVR engagements, sensor performance and cockpit displays must be well represented and models of sensors, missiles, and the EW environment must be credible.

Combat within visual range (WVR), on the other hand, is usually restricted to one versus a few players and usually requires higher fidelity out-of-the-window visuals. Whilst the level of tactical teaming and C3 complexity for a computer generated entity does not therefore need to be as high, the behaviour must still be realistic and representative of human pilots.

Most tactics developed in AOD for use with “intelligent” agents are for BVR scenarios, as this has been required for RAAF tasking. Tactics for WVR engagements are usually scripted rather than being agent controlled. In this Technology Demonstrator we are trying to span the range from BVR to WVR with intelligent agents (CGFs) interacting with a HiL.

11. Future Work

The initial experiments described in this report have demonstrated the feasibility of linking Computer Generated Forces (via BattleModel) with a Human-in-the-Loop simulation (via the AOSC). Much more can be achieved with such a system, as described in the introduction. In order to accomplish this, there are several areas which need immediate development and others which are slightly longer term. These are outlined below.

11.1 Immediate Development Work

The feasibility of linking CGFs and a HiL simulation has been demonstrated. However, in order to demonstrate that the CGFs can be endowed with more intelligence than they have exhibited thus far, the issues identified in the discussion (section 10) need to first be addressed. These include:

- More realistic sampling intervals being implemented within the BattleModel DIS interface (i.e. increase sampling for entity state PDUs from 1 per second to 20 per second);
- CGFs being able to pursue more than one entity after achieving radar lock and killing opponent;
- CGFs having variable speed in order to pursue the HiL;
- CGFs returning to initial role as patrol once engagement has been completed;
- Orange CGFs being able to fly CAP (Combat Air Patrol).

Then, in priority order, the following issues need to be addressed:

- Missile flyout model;
- Improvement of the BattleModel DIS interface (including the ability to incorporate more than one external entity and to handle emission PDUs);
- Agent development (including teamwork and voice communication); and
- Database matchups.

These points are expanded upon in section 11.3.

Possible scenarios that would be useful to investigate will now be outlined. This will help to determine the further developments that are required.

11.2 Phased Plan for Scenario Development

It is envisaged that the existing research testbed, with the modifications outlined above, will be able to be used to develop (over time) a full Technology Demonstrator. In order to demonstrate the usefulness of combining the BattleModel with a Human-in-the-Loop simulation facility, a number of scenarios are proposed, each increasing in complexity. It should be noted, however, that all the scenarios involve only air to air engagements since the BattleModel does not have an interface to terrain. The incorporation of terrain model development in the BattleModel would be desirable.

1v1 Scenario: 1 CGF v 1 HiL

The CGF would fly a very simple intercept (pure pursuit against the HiL). The purpose and focus of this scenario would be to test newly implemented interface models. The capabilities of both the HiL and the CGF would be:

- Use of radar (they will detect and track each other); and
- Use of air-to-air missiles (they can kill each other).

2v1 Scenario: 2 CGFs v 1 HiL

Existing team tactics would be used for the CGFs, which would fly as a pair. The tactics for the CGFs currently exist (within AOD) and would not need to be developed. The purpose of this scenario would be to test the interface model(s) for a multi-aircraft side and demonstrate the existing team tactics in the environment.

2v2 Scenario: 2 CGFs v 1 CGF + 1 HiL

The side with two CGFs would be as in the previous scenario. The HiL would be the wingman in the pair on the other side. The CGF leader would perform *all* team decisions (sorts targets, selects intercept tactic), and give the leader instructions about what to do at the team level. This would require a communications mechanism from the CGF leader to the HiL wingman (but not from the HiL to CGF). This communication is normally by voice and would involve the use of a text to voice translator.

2v2 Scenario: 1 CGF + 1 HiL v 1 CGF + 1 HiL

The purpose of this scenario would be to demonstrate the capability of multiple HiLs. The fact that the HiLs are on different teams eliminates the need for voice communication between the HiLs. An option would be to have both HiLs on the same side, or, alternatively to have three HiLs, two on one side and one HiL wingman on the other.

8v4 Scenario: 8 CGFs v 3 CGFs + 1 HiL

The purpose of this scenario would be to demonstrate the scalability of the system. The eight CGFs would consist of two teams, one four-ship sweep team, and one team of two strikes and two escorts. The tactics for these CGFs already exist within AOD (in a classified form). The other side would consist of two teams, each one a pair flying a combat air patrol (CAP) in a defensive counter air role. They would be defending a high value asset and would respond to any air threat by trying to neutralise it. The scenario could be scaled up as required. The only limitation is that the HiL must be a wingman as there is currently no voice communication to enable a pilot leader to fly with a computer generated wingman. It would also be possible to use two or more HiLs instead of just one, as in the previous scenario.

HiL leader and CGF wingman development

The main limitation for any more complex scenarios is that interaction between HiL leaders and CGF wingmen would require voice recognition software by the CGF. This requirement was eliminated in the above scenarios by making the HiL the wingman so that the pilot just follows orders. If the HiL is to be anything other than a wingman (or a number 4 in a four ship) it will be required to give orders to its wingman, which is normally (and so should be here) done by voice.

An operational HiL leader with a CGF wingman would represent a very significant step, and is a viable goal considering the defined semantics of fighter radio communications. Some voice recognition work has already been achieved within the AOSC, and this Technology Demonstrator would be able to build upon existing expertise.

11.3 Model Development

11.3.1 Reasoning Models (Agent Development)

Further development of this Technology Demonstrator as outlined by the scenarios in section 11.2 will concentrate on incrementally increasing the complexity of the tactics and the level of team interaction, initially within the CGFs, and then involving the HiL (or Humans-in-the-Loop). Tactics for a range of aircraft types acting in a range of roles are available. These standard tactics are not designed with HiL interaction in mind and there may be some additional work involved in modifying these plans to enable interaction with HiL players.

In relation to the scenarios of section 11.2, the first step would be to investigate the team level interaction in a pair, with the HiL simulation acting in the role of either a wingman or a leader. The HiL wingman case would be easier because the wingman's actions are generally either independent, or in response to a command from the leader. A text to voice converter attached to the CGF leader can issue commands to the HiL.

The HiL leader is a more difficult case experimentally, because it requires the CGF to understand verbal orders from the leader. This could be done using voice recognition software or by sending data using simulated data links, although this would not be operationally realistic (voice communications are used most of the time).

In addition to the work described above at the pair level, some work is required in the following more general areas of agent development (to increase fidelity):

- expansion of the tactics suite (more aircraft types, more roles, BVR, WVR);
- recognition of intention;
- better representation of team work and the organisational structure; and
- better representation of human behaviour (at the level of individual pilot skill, considering factors such as pilot reaction to increased workload).

11.3.2 Interface Models

The interface models developed thus far have been successful for the research testbed (Phase One of the Technology Demonstrator). However, there are several aspects which will result in an improvement to the models, namely:

- Removal of the hard-wired flags and values in the current AOSCtoBM and BMtoAOSC models, in order to make the code more flexible. This includes the ability to select sensible default settings. The PDU sample rate can be altered here;
- Modification of platform type mapping in the interface model, so that it more closely follows the DIS protocols;
- Utilisation of the existing interface model to generate and process BattleModel missile data, both in the form of DIS PDU's and BattleModel data. This function is already present.

- Implementation of Electronic Warfare (via Emission PDUs) into the interface model. An appropriate model will be needed to produce and send out such data. The information in the emission PDUs is used to represent radar emissions and would allow the modelling of radar and missile usage.
- Development of the BattleModel and its interface to handle more than one external PDU. The present AOSctoBM model handles 3 PDUs within the one model. It is proposed to break this down into three separate models. This same methodology can then be used to scale up the number of external entities interacting with the BattleModel.

11.3.3 Physical Models

Work is also required on physical models in the BattleModel.

A missile flyout model is essential. The currently implemented missile is a percentage kill model; the position of the missile is not calculated at all. A flyout model is required so that the position of any fired missiles can be sent to the AOSC to be displayed. There is also a corresponding requirement to implement a missile flyout component of the BattleModel, and then correctly implement the DIS procedure of having the attacked simulator (e.g. HiL) make the decision on whether a kill was effective or not. DIS protocol dictates that it is the receiving entity which decides whether it is killed or damaged. The BattleModel should not make the decision on a kill, when attacking the HiL simulator.

Terrain model match-ups between AOSC visual/terrain databases and the BattleModel database need to be examined.

11.4 Future Development Work

dGame is the successor to BattleModel, and will have a distributed architecture, which BattleModel does not. It is currently being developed by AAIL in collaboration with AOD for operations analysis studies. Because of its distributed framework it lends itself more readily to being interfaced with one or more Human-in-the-Loop simulations, allowing real-time interaction of human and intelligent computer generated forces. It is envisaged that dGame together with a Human-in-the-Loop facility will be able to provide a basis for simulation-based training and mission rehearsal systems.

HLA (High Level Architecture) is replacing DIS as the international networking standard for simulation. It also replaces the ALSP (Aggregate Level Simulation Protocol) used to join constructive simulations (the BattleModel is a constructive simulation). HLA allows virtual (i.e. HiL-type simulations) and constructive simulations to interact with each other more readily. Since all United States simulations have been mandated to use HLA as the networking architecture from 2000 onwards it would be prudent to investigate the use of HLA for interoperability if required.

Consequently, it is envisaged that HLA will be used as the networking architecture for the BattleModel and Human-in-the-Loop interaction, at a later date.

A fully functional system based on this Technology Demonstrator would provide another method for the verification and validation of CGF tactics, especially when flying against human opponents. In addition, it will provide the ability to capture novel tactics and innovative behaviour from human experts by using different arrangements of CGFs allowing a variety of non-standard scenarios to be explored.

12. Conclusions

Phase One of the Technology Demonstrator has been the establishment of a research testbed, in which we have successfully linked the BattleModel with the AOOSC and exchanged Entity State PDUs. Connectivity of the AOOSC and the BattleModel with a simple scenario and simple tactics has been achieved using DIS networking protocols. This demonstrated that CGFs and a HiL can, and do, interact in a real-time combat situation. The demonstration scenario evolved as expected with orange and blue CGFs engaging each other and the HiL.

This research testbed can now be used to further investigate the interaction of CGFs (via the BattleModel) and a HiL facility (for example, the AOOSC) in the same synthetic environment. This will enable the construction of a full Technology Demonstrator which will demonstrate its usefulness by providing a source of suitable, intelligent CGFs to interact with Human-in-the-Loop facilities.

Further experimentation is needed to determine how realistic the interactions of HiLs and CGFs are in air engagements and from this to determine how best to utilise this capability. More complex scenarios are required to test both BattleModel tactics and human reactions. To achieve this, further development is required to produce more realistic models (physical (e.g. missile flyout), tactical (e.g. teamwork), reasoning, and interface). The agents also need development in several areas to provide more realistic HiL-CGF interactions.

Utilisation of the High Level Architecture (HLA) and dGame will be explored at a later development stage. The experience gained in linking the BattleModel with the AOOSC will allow similar linkages to dGame, perhaps using HLA.

13. References

- [1] DIS Steering Committee "The DIS Vision: A Map to the Future of Distributed Simulation", Technical Report, IST-ST-94-01, Institute for Simulation and Training, University of Central Florida, 1994.

- [2] Pongracic H., Appla D., and McIlroy D., "Are Intelligent Computer Generated Forces Now Available for Human-in-the-Loop Simulations?". In: Proceedings of the Simulation Technology and Training Conference, SimTecT 97, Canberra, Australia, 1997, pp 85-90.
- [3] Lloyd I., BattleModel Overview, July 1999, unpublished.
- [4] Murray G., Steuart S., Appla D., McIlroy D., Heinze C., Cross M., Chandran A., Raszka R., Tidhar G., Rao A., Pegler A., Morley D. and Busetta P., "The Challenge of Whole Air Mission Modelling". In: Proceedings of the Australian Joint Conference on Artificial Intelligence Workshop on AI in Defence, Canberra, Australia, S. Goss (Ed.), 1995.
- [5] Ryan P.J., and Clark P.D., "Distributed Interactive Simulation for the Royal Australian Navy", DSTO Client Report DSTO-CR-0075, October 1997.
- [6] Rao A., Morley D., Selvestrel M. and Murray G., "Representation, Selection, and Execution of Team Tactics in Air Combat Modelling". In: Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, A. Adams and L Sterling (Eds), World Scientific, November 1992, pp 185-190.
- [7] Tidhar G., "Organisation-Oriented Systems: Theory and Practice", PhD Thesis, Department of Computer Science, The University of Melbourne, 1998.
- [8] Tidhar G., Selvestrel M., and Heinze C., "Modelling Teams and Team Tactics in Whole Air Mission Modelling". In: Proceedings of the Eighth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, (IEA/AIE-95), Forsyth, G.F. and Ali, M. (Eds), Melbourne, Australia, 1995, pp 373-381.
- [9] Rao A. S., Lucas A., Morley D., Selvestrel M., Murray G., "Agent-Oriented Architecture for Air-Combat Simulation", Technical Note 42, The Australian Artificial Intelligence Institute, 1993.
- [10] McIlroy D., Heinze C., "Air Combat Tactics Implementation in the Smart Whole AiR Mission Model (SWARMM)". In: Proceedings of The Simulation Technology and Training Conference, SimTecT 96, Melbourne, Australia, 1996.
- [11] McIlroy D., Heinze C., Appla D., Busetta P., Tidhar G., Rao A., "Towards Credible Computer-Generated Forces". In: Proceedings of The Simulation Technology and Training Conference, SimTecT 97, Canberra, Australia, 1997.

- [12] Tidhar G., Murray G., Steuart S., "Computer Generated Forces and Agent Oriented Technology". In: Proceedings of the Australian Joint Conference on Artificial Intelligence Workshop on AI in Defence, Canberra, Australia, S. Goss (Ed), 1995.
- [13] Heinze C., Smith B., and Cross M., "Thinking Quickly: Agents for Modeling Air Warfare". In: Proceedings of the Australian Joint Conference on Artificial Intelligence AI '98, Brisbane, Australia, 1998.
- [14] Clark, P., Pongracic, H., Cross, M., and Morton, J., "Incorporating Intelligent CGF in a Human-in-the-Loop Simulation". In: Proceedings of the Fourth International SimTecT Conference, SimTecT 99, Melbourne, Australia, 1999, pp 205-210.
- [15] Goss S., personal communication 1999.
- [16] "VR-Link: The Virtual Reality Networking Toolkit" Release 2.4.6. Technologies Inc, 1997.

14. Acknowledgements

The authors wish to thank Mr James Briggs and Mr Arvind Chandran for their assistance with the development of BattleModel tactics and models based on the AOSC scenario, as well as installation of the BattleModel within the AOSC's SGI computers. The authors also wish to thank Mr David Creed for his assistance during the experimental setup.

The assistance of the Australian Artificial Intelligence Institute (AAIL) in enabling the progression of the development of this technology demonstrator is appreciated.

Appendix A: BattleModel DIS Interface Programs

A.1. Introduction

Two programs have been written to implement the BattleModel-AOSC interface using DIS. The AOSC is already DIS compliant, so an interface was required to the BattleModel which provides PDUs from the BattleModel for use in the AOSC and converts PDUs coming from the AOSC to an appropriate form for use by the BattleModel. These programs are written in C++ and incorporate classes and functions from the VR-Link toolkit. The VR-Link toolkit defines a set of classes (and functions) for manipulating PDUs. The user must write code which incorporates these classes in some way. VR-Link version 2.4.6 was used.

The first of these programs, *Receive_PDUs* (called AOSCtoBM in the main paper), processes PDUs from the network to provide information that is used within BattleModel. The second program, *PackageAndSend_PDUs* (called BMtoAOSC in the main paper), takes information provided by BattleModel and constructs, then broadcasts, appropriate DIS PDUs to the network. Three PDU types are currently implemented; Entity State PDU (ESPDU), Fire PDU (FPDU) and Detonation PDU (DPDU). The classes and functions from the VR-Link toolkit are described in "VR-Link: The Virtual Reality Networking Toolkit" [16].

Different coordinate systems are used by the BattleModel and the AOSC. The BattleModel has a geodetic coordinate system, whereas the DIS protocol uses a geocentric system. The interface programs perform the conversion between coordinate systems as required.

A.2. Program Description

VR-Link classes and functions are denoted as **boldface**. User defined code is denoted by *italics*.

A.2.1 Overview

The class *Battlemodel_Interface_Data* contains the data that is passed between the BattleModel and the DIS network (to which the AOSC is connected). *Receive_PDUs* takes data from incoming PDUs, converts it as necessary then writes it to *Battlemodel_Interface_Data* for use by the BattleModel. *PackageAndSend_PDUs* takes data from *Battlemodel_Interface_Data* that were placed there by the BattleModel, converts it as necessary, then creates the relevant PDU types. These PDUs are then broadcast onto the DIS network.

A.2.2 Receive_PDUs

A2.2.1 Entity State Protocol Data Unit (ESPDU)

In *Receive_PDUs*, the class **drainInput** (which reads information from the network) will go through the Entity State PDUs from the network queued at **DtExerciseConn**, (the class which establishes and maintains a network connection). The class **DtRva** maintains a list of entities in an exercise. Subclassing **DtRva** and redefining the virtual functions **acceptPdu**, **entityAdded** and **removeAndDelete** allow for Entity StatePDU filtering, interception of new entities, and interception before entity deletion, respectively.

A2.2.1.2 Filtering

DtExerciseConn will filter out all PDU types in which it has no interest and keep a track of all PDU types in which it is interested. By default those PDU types with a different **exerciseId** will be filtered out. The user nominates the PDU types in which he/she is interested. For the initial connectivity of the BattleModel and the AOSC, only the Entity State PDU, Fire PDU and Detonate PDU are required; all other PDU types are filtered out. The function **acceptPdu** filters out on **site_id** and **application_no.** and has been coded to reject Entity StatePDUs issued by *PackageAndSend_PDUs*.

A2.2.1.3 New Entities

When an Entity State PDU is received for an entity not already on the **DtRva**, **entityAdded** calls "addToMyOwnEntityList" which updates the structures *DIS_Aircraft_Data* and *DIS_Missile_Data*, (these are referred to as "myOwnEntityList"). This is coded this way since *Receive_PDUs* cannot directly access the *Battlemodel_Interface_Data* from routines subclassed off the **DtRva**.

A2.2.1.4 Removing Entities

If an Entity StatePDU with the "final appearance" flag set to true is received, or no Entity StatePDU for an entity is received after a timeout period, then **DtRva** will call the function **removeAndDelete**. This calls *removeFromMyOwnEntityList* which will obtain the final position, velocity and orientation data before the entity is deleted. Even though an entity is to be removed from the exercise, the final position, velocity and orientation must still be set in "myOwnEntityList" in order to inform the BattleModel.

In *Receive_PDUs*, the code for "process new or deleted entities", will first copy data from the updated "myOwnEntityList" to *Battlemodel_Interface_Data* so that BattleModel will have a copy of information from deleted entities. It is not possible to access the *Battlemodel_Interface_Data* from the virtual functions subclassed off the **DtRva**. Hence, it is necessary to have this first step where new or deleted entity data is copied from "myOwnEntityList" to the *Battlemodel_Interface_Data*. The *activeInExerciseFlag* is set

here. If a Fire PDU was received for a missile before the first Entity State PDU for that missile was received, then details from the Fire PDU would have been set, but the *activeInExerciseFlag* is only set by the arrival of the first Entity State PDU.

In *Receive_PDUs*, within the section of code "process entities on remote entity list", entities in *Battlemodel_Interface_Data* which have *activeInExerciseFlag* true and *removeFromExerciseFlag* false will have position, velocity and orientation converted, then copied, from the entity list on *DtRva* into *Battlemodel_Interface_Data*.

A2.2.2 Fire Protocol Data Unit (FPDU)

(i) A callback for Fire PDUs is defined in *main()*. A callback function for a particular PDU type is a function which is called whenever that particular type of PDU is encountered. (Callbacks for Entity State PDUs are done automatically by VR-Link, but callbacks for Fire PDUs and Detonate PDUs need to be user specified.) Therefore, in *drainInput*, any Fire PDUs (FPDUs) will be processed by a function called *fpdub*.

(ii) This code will:

- reject Fire PDUs for non-trackable munitions;
- reject Fire PDUs from *PackageAndSend_PDUs* code; and
- update the *missileStatusFlag* indicating that a Fire PDU has arrived.

(iii) If an Entity State PDU has already been received for this Fire PDU then the fields *launchId*, *targetId* and *range* will be set. If the Fire PDU arrived before the first Entity State PDU for that entity, then the *entityId* field will also be set.

A2.2.3 Detonate Protocol Data Unit (DPDU)

(i) A callback for Detonate PDUs is defined in *main()*. Therefore, in *drainInput*, any Detonate PDUs (DPDUs) will be processed by a function called *dpdub*.

(ii) This code will:

- reject Detonate PDUs for non-trackable munitions;
- reject Detonate PDUs from *PackageAndSend_PDUs* code; and
- update the *missileStatusFlag* indicating that a Detonate PDU has arrived.

A.2.3 PackageAndSend_PDUs

In *PackageAndSend_PDUs*, the section of code "process BM aircraft" converts position, velocity and orientation from the BattleModel coordinate representation to that used by the DIS protocols. *DtTick*, a class that determines if a threshold for Entity State PDU update has been reached, is used to issue Entity State PDUs. If an aircraft is to be removed from the exercise the "final appearance" flag is first set in the Entity State PDU.

In *PackageAndSend_PDUs*, the section of code "process BM missiles" converts position and velocity from the BattleModel coordinate representation to that used by the DIS protocols. A Fire PDU or Detonate PDU is issued depending on flag settings. *DtTick* is used to issue Entity State PDUs. If a missile is to be removed from the exercise the "final appearance flag" is first set in the Entity State PDU.

A.2.4 Flags used to pass status information within programs

Within the *Battlemodel_Interface_Data* class the *Aircraft_Data* structure uses an *activeInExerciseFlag* and a *removeFromExerciseFlag*. The *Missile_Data* structure uses these two flags as well as a *missileStatusFlag*. These flags are used differently by *Receive_PDUs* and *PackageAndSend_PDUs*.

In these interface programs (*Receive_PDUs* and *PackageAndSend_PDUs*) *Battlemodel_Interface_Data* is used to pass information between the BattleModel and the AOSC simulations. The number of aircraft and the number of missiles in the exercise are fixed and defined at initialisation. The *activeInExerciseFlag* indicates when an aircraft or missile is taking part in the exercise. The *removeFromExerciseFlag* indicates when an aircraft or missile no longer takes part in an exercise. Once turned on, the flags *activeInExerciseFlag* and *removeFromExerciseFlag* are never turned off in the exercise.

A2.4.1 Receive_PDUs

The *activeInExerciseFlag* is set to true when the first Entity State PDU for that entity is processed. The *removeFromExerciseFlag* is set to true when the "final appearance" flag in an Entity State PDU is set, or when no Entity State PDU for an entity is received before the timeout period has elapsed. These flags are used internally by *Receive_PDUs* to determine appropriate processing and are also passed as information to the BattleModel.

If a Fire PDU is received, the *fpdub* code will set the *missileStatusFlag* to the value "1". If the *activeInExerciseFlag* for this missile has also been set then these two flags inform BattleModel that a missile has been fired and is active in the exercise. If the *activeInExerciseFlag* for this missile has not been set then the Fire PDU information updates the missile data, but the missile will not yet take part in the exercise.

If a Detonate PDU is received, the *dpdub* code will set the *missileStatusFlag* to the value "2". This will inform BattleModel that the missile has detonated.

A2.4.2 PackageAndSend_PDUs

The *activeInExerciseFlag* is set to "1" by BattleModel to indicate that the aircraft or missile should be processed by the DIS interface. The *removeFromExerciseFlag* is set to "0" at initialisation. BattleModel will set it to "1" if the aircraft or missile is to be removed from the exercise. The *PackageAndSend_PDUs* program will set the final appearance flag in the Entity State PDU for this aircraft or missile and set the *removeFromExerciseFlag* to "2" to prevent further processing. The *missileStatusFlag* is initialised to "0" to indicate no action. BattleModel will set it to "1" if a Fire PDU is to be issued for the missile, or set it to "2" if a Detonate PDU is to be issued. *PackageAndSend_PDUs* will set the *missileStatusFlag* to "3" after it has sent the Fire PDU or Detonate PDU.

A.2.5 Synchronisation when PDUs are out of order

It is expected that under normal conditions a Fire PDU will arrive just before the first Entity State PDU for a missile, and a Detonate PDU will arrive just before an Entity State PDU with the "final appearance" flag set, for a particular missile. However, due to network delays it is possible for such a Fire PDU or Detonate PDU to arrive later than their corresponding Entity State PDU. *Receive_PDUs* has been coded to manage each of these situations.

Only the arrival of the first Entity State PDU for a missile will result in the *activeInExerciseFlag* being set to true. The arrival of the Fire PDU will merely set the *launcherId* and *targetId*. The arrival of a Detonate PDU will cause the *missileStatusFlag* to be set to "2". The arrival of the corresponding Entity State PDU with the final appearance flag set will cause the *removeFromExerciseFlag* to be set. Either of these last two flags (*missileStatusFlag* and *removeFromExerciseFlag*) will allow BattleModel to decide that the missile has detonated.

DISTRIBUTION LIST

Incorporating Intelligent Computer Generated Forces in a Human-in-the-Loop Simulation Phase One - Establishment of a Research Testbed

Peter Clark, Helen Pongracic, Martin Cross,
Jim Morton, and John Fulton

AUSTRALIA

DEFENCE ORGANISATION

Task Sponsor

ER&D

S&T Program

Chief Defence Scientist	} shared copy
FAS Science Policy	
AS Science Corporate Management	
Director General Science Policy Development	
Counsellor Defence Science, London (Doc Data Sheet)	
Counsellor Defence Science, Washington (Doc Data Sheet)	
Scientific Adviser to MRDC Thailand (Doc Data Sheet)	
Scientific Adviser Policy and Command	
Navy Scientific Adviser (Doc Data Sheet and distribution list only)	
Scientific Adviser - Army (Doc Data Sheet and distribution list only)	
Air Force Scientific Adviser	
Director Trials	

Aeronautical and Maritime Research Laboratory

Director

Chief of Air Operations Division	Dr Helen Pongracic
Dr Simon Oldfield, RL-SHF	Mr Martin Cross
Dr David Glenney, RL-AOA	Dr Jim Morton
Dr Mather Mason, HST	Mr John Fulton
(Task Manager DST 98/148)	Mr James Briggs
Dr Robin Miller, HTS	Mr David McIlroy
Dr Gavan Lintern, HHF	Mr David Creed
Mr John Harvey, HAOSC	Dr Peter Ryan
Mr Terry Kinsella, HSA	Dr Lucien Zalcman
Mr Ian Lloyd, HCMA	Dr Simon Goss
Mr Graeme Murray, HAOA	Mr Dino Appia
Ms Serena Steuart, HCAO	Mr Arvind Chandran
Dr Peter Clark (11 copies)	Mr Clint Heinze

DSTO Library and Archives

Library Fishermans Bend
Library Maribyrnong
Library Salisbury (2 copies)
Australian Archives
Library, MOD, Pyrmont (Doc Data sheet only)
*US Defense Technical Information Center, 2 copies
*UK Defence Research Information Centre, 2 copies
*Canada Defence Scientific Information Service, 1 copy
*NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

Capability Development Division

Director General Maritime Development (Doc Data Sheet only)
Director General Land Development (Doc Data Sheet only)
Director General C3I Development (Doc Data Sheet only)
Director General Aerospace Development
Head Australian Defence Simulation Office

Navy

SO (Science), Director of Naval Warfare, Maritime Headquarters Annex,
Garden Island, NSW 2000. (Doc Data Sheet only)

Army

ABCA Standardisation Officer, Puckapunyal, (4 copies)
SO (Science), DJFHQ(L), MILPO Enoggera, Queensland 4051 (Doc Data Sheet only)
NAPOC QWG Engineer NBCD c/- DENGSR-A, HQ Engineer Centre Liverpool
Military Area, NSW 2174 (Doc Data Sheet only)

Intelligence Program

DGSTA Defence Intelligence Organisation
Defence Intelligence Organisation-Information Centre

Corporate Support Program

OIC TRS, Defence Regional Library, Canberra

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy
Library
Head of Aerospace and Mechanical Engineering
Serials Section (M list), Deakin University Library, Geelong, 3217
Senior Librarian, Hargrave Library, Monash University
Librarian, Flinders University

OTHER ORGANISATIONS

Dr Mike Georgeff, Director Australian Artificial Intelligence Institute
Dr Jeremy Samuel, AAIL
NASA (Canberra)
AGPS

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS

Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

INFORMATION EXCHANGE AGREEMENT PARTNERS

Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US
National Aerospace Laboratory, Japan
National Aerospace Laboratory, Netherlands

SPARES (10 copies)

Total number of copies: 91

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA					
				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
2. TITLE Incorporating Intelligent Computer Generated Forces in a Human-in-the-Loop Simulation: Phase One - Establishment of a Research Testbed			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)		
4. AUTHOR(S) Peter Clark, Helen Pongracic, Martin Cross, Jim Morton, and John Fulton			5. CORPORATE AUTHOR Aeronautical and Maritime Research Laboratory PO Box 4331 Melbourne Vic 3001 Australia		
6a. DSTO NUMBER DSTO-TR-0889		6b. AR NUMBER AR-011-122		7. DOCUMENT DATE October 1999	
8. FILE NUMBER M1/9/635		9. TASK NUMBER DST 98/148		10. TASK SPONSOR ER&D, AOD	
				11. NO. OF PAGES 34	
				12. NO. OF REFERENCES 16	
13. DOWNGRADING/DELIMITING INSTRUCTIONS			14. RELEASE AUTHORITY Chief, Air Operations Division		
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT <i>Approved for public release</i>					
OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108					
16. DELIBERATE ANNOUNCEMENT No Limitations					
17. CASUAL ANNOUNCEMENT Yes					
18. DEFTEST DESCRIPTORS Simulation, Artificial Intelligence, Distributed Interactive Simulation					
19. ABSTRACT The Air Operations Division (AOD) of the Defence Science and Technology Organisation (DSTO) has developed a range of simulation tools whose capabilities include air combat modelling. The Air Operations Simulation Centre (AOSC) provides a facility for Human-in-the-Loop (HiL) simulation, and the BattleModel provides intelligent Computer Generated Forces (CGFs). The AOSC and the BattleModel have been integrated to create a research testbed for air combat simulation as Phase One of a Technology Demonstrator. This has been the first integration of operationally credible CGFs with a HiL facility in an air environment by DSTO. The CGFs have been implemented using dMARSTM agent controlled fighter aircraft. Interaction with the AOSC has been achieved using Distributed Interactive Simulation (DIS). This paper outlines the rationale for using intelligent CGFs and HiLs in the same synthetic environment, details the development of the research testbed, describes the initial results achieved and makes suggestions for the way forward.					